

Analyse et amélioration de la maturité du code d'une application

Steve Scherz

Contexte

La qualité d'un logiciel dépend de nombreux facteurs, tels que la couverture des besoins métiers, la robustesse du code ou les possibilités d'évolutions et de maintenance. Dans de nombreux projets, les pressions liées au temps et aux ressources font que la priorité est mise sur la satisfaction des besoins. La robustesse et la maintenabilité du code passent en second plan.

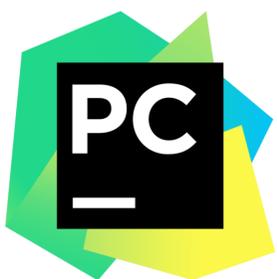
Objectifs

Réaliser un état de l'art sur le sujet de la maturité du code. Cela doit servir pour réaliser une analyse d'une application développer par l'HEG-arc pour la fédération suisse de boxe light-contact.

Identifier et rédiger de nouveaux tests pour améliorer leurs couvertures du code sur la base des résultats obtenus lors de la réalisation de l'objectif précédent

Proposer des recommandations sur le processus de développement pouvant améliorer la maturité du code source.

Technologies et langages



docker



Pourquoi écrire un beau code ?

- Facilite la collaboration et l'échange
- Améliore la maintenabilité
- Réduit les couts d'entretien
- Facilite les évolutions

Démarche



Recherche documentaire sur le sujet de la maturité du code



Analyse de la qualité du code source, des tests déjà existants et détermination des tests à ajouter



Rédaction des nouveaux tests identifier précédemment et correction du code de l'application



Supervision d'évolution de la maturité du code par l'intermédiaire d'outils de tests continu

Résultats

- Progression du nombre de tests de 64 à 234
- Augmentation de 57 % à 74 % de la couverture
- Plus de 3000 nouvelles lignes de code